## REMARKS

Claims 1-9 are pending. Claims 3 and 5 were amended to address the rejection under 35 U.S.C. § 112, second paragraph, rejection. Clams 7 and 9 were amended solely to improve their form.

For at least the reasons set forth below, withdrawal of all outstanding objections and rejections is respectfully requested.

### 35 U.S.C. § 112, second paragraph, rejection

Claims 3 and 5 were amended to clarify that the transactions are "multi-threaded," as described in the specification. See, page 6, lines 5-7 and 12-15; and page 24, lines 9-12. In view of the amendment, withdrawal of this rejection is respectfully requested.

### Double Patenting Rejection

Although Applicants disagree with the double patenting rejection, to advance prosecution of the present application, Applicants submit a Terminal Disclaimer under 37 C.F.R. 1.321(b) herewith, stating that the '196 patent and the present application are commonly owned and disclaiming the terminal part of the statutory term of any patent granted on the present application which would extend beyond the full statutory term of the '196 patent.

Likewise, although Applicants disagree with the double patenting rejection, to advance prosecution of the present application, Applicants submit a Terminal Disclaimer under 37 C.F.R. 1.321(b) herewith, stating that the '129 patent application and the present application are commonly owned and disclaiming the terminal part of the statutory term of any patent granted on the present application which would extend beyond the full statutory term of any patent issuing based on the '129 patent application.

In view of the accompanying Terminal Disclaimer which includes both the '196 patent and the '129 patent application, withdrawal of the double patenting rejection is respectfully requested.

**Prior Art Rejections**

Claims 1, 2 and 4 were rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by U.S. Patent No. 5,799,305 (Bortvedt et al.), hereafter, "Bortvedt."

Claims 6-9 were rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by te symposium paper authored by Holliday et al., hereafter, "Holliday."

Applicants respectfully traverse both of these rejections.

1. Patentability of claims 1 and 4 over Bortvedt

Bortvedt discloses a method for determining commitment of a distributed transaction in a distributed database system. The distributed transaction includes an owner and a helper. The method operates as follows:

1. An interval coordinator and a plurality of coservers are provided.
2. The owner associates with a first coserver, the helper associates with a second coserver, and each of the coservers are associated with a transaction log.
3. A succession of interval messages are sent from the interval coordinator to each of the coservers. The interval messages represent a succession of temporal periods.
4. The transaction log associated with the coserver is flushed to non-volatile storage in response to receiving one of the interval messages.
5. A state is maintained in each of the coservers that identifies a most recently received interval message.
6. A closure message is transmitted from each of the coservers to the interval coordinator after that coserver flushes its associated transaction log.
7. A request message is transmitted from the owner to the helper identifying an operation in the distributed transaction for the second coserver to execute.
8. A completion message is transmitted from the helper to the owner upon execution of the operation. The completion message includes a tag identifying the most recently received interval message of the second coserver.
9. After receiving the completion message, an eligibility message is transmitted for the transaction from the owner to the interval coordinator.

10. After receiving the eligibility message from the owner and a closure message from the helper, a commit state is written for the transaction to stable storage.
11. After writing the commit state, a commit message for the transaction is sent from the interval coordinator to the owner and helper.

In the outstanding Office Action, the Examiner asserts that the "interval message" in Bortvedt is equivalent to the claimed "ready to commit token." Applicants respectfully disagree with this position. "Ready to commit tokens" are assigned to transactions, not time intervals, as shown in the following claim excerpts (underlining added for emphasis):

> Claim 1
> (ii) assigning a ready to commit token to the transaction
>
> Claim 4
> (b) a replication engine at the first node assigning a ready to commit token to the transaction in coordination with the application

In contrast to the claimed invention, and as also described above, the interval message in Bortvedt represents a succession of temporal periods. See, also, the following text portions of Bortvedt (underlining added for emphasis):

> As discussed below and as shown in FIG. 9, database system 5 generates a regular exchange of messages between IC 110 and IPs 115a-115g. By way of example, database system 5 includes seven coservers 102a-102g, but there can be a different number of coservers, as needed for a particular application. At the beginning of each interval, IC 110 transmits an "interval message" 120 to every IP 115a-115g. Interval message 120 informs IPs 115a-115g that a new interval has commenced. In a preferred embodiment, IC 110 transmits interval message 120 about every one-hundred milliseconds. The length of time between intervals will vary with different configurations, but preferably should be longer than the time required to send and receive a message and to flush a page to a transaction log.

Although there may be transaction information in the interval message (e.g., see Fig. 12), the interval message is not assigned to a transaction. In fact, different interval messages may contain similar transaction information.

The Examiner further asserts that the "completion message" is equivalent to the sent back message in the following claimed steps (underlining added for emphasis):

> Claim 1
> (iv) determining at the one or more other nodes whether the respective databases are prepared for a commit operation for the transaction corresponding to the ready to commit token, and, if so, sending back the ready to commit token to the originating node
>
> Claim 4
> (d) a replication engine at a second node determining whether a target database at the second node is prepared for a commit operation for the transaction corresponding to the ready to commit token, and, if so, sending back the ready to commit token to the first node

Applicants also respectfully disagree with the Examiner's analogy. The "completion message" is merely a message that is transmitted from the helper to the owner upon execution of an operation, and it includes a tag identifying the most recently received interval message of the second coserver (see step 8 above). Since there is nothing equivalent to a ready to commit token in Bortvedt, there can be no step of sending it back, either directly or indirectly as a different message. Second, since the request message in Bortvedt is not a message that is associated with a specific transaction, the completion message has nothing whatsoever to do with whether a database is prepared for a commit operation for a transaction associated with an originally sent request message.

In sum, Bortvedt does not disclose or suggest the claimed "ready to commit" token, or any equivalent message that performs an analogous function to the "ready to commit" token. Claims 1 and 4 are thus believed to be patentable over Bortvedt.

2. Patentability of claim 6 over Holliday

Claim 6 reads as follows (underlining added for emphasis):

> 6. A method of performing dual writes for replicating transactions among plural databases located at different nodes, each transaction being one or more transaction steps or transaction operations, at least some of the

transaction steps or transaction operations being update steps or operations, the method comprising:
(a) initiating a transaction at an originating node;
(b) inhibiting the dual write replication process from communicating transaction steps or operations of the transaction with one or more other nodes until an update step or operation occurs within the transaction at the originating node; and
(c) upon the occurrence of the update step or operation, performing the dual write replication process with respect to the one or more other nodes and sending with the update step or operation all transaction steps or operations for the transaction that have occurred prior to the update step or operation for the transaction, or prior to the previous update step or operation if a previous update step or operation existed for the transaction.

In the outstanding Office Action, the Examiner asserts that step (c) is disclosed by Protocol A3 (delayed broadcast writes) described on page 160, column 1 of Holliday. Applicants respectfully disagree. Protocol A3 of Holliday does not perform dual write replication upon the occurrence of an update step or operation. In fact, Protocol A3 of Holliday teaches against this limitation because update operations are deferred until commit time, when a single message with all updates is sent to all other sites. Holliday further explains that "a write operation is deferred until $T_i$ is ready to commit." Thus, in Protocol A3 of Holliday, no dual write replication is performed upon the occurrence of an update step or operation. At best, Protocol A3 of Holliday discloses performing dual write replication upon the occurrence of a "commit."

Since Protocol A3 of Holliday does not disclose or suggest the first portion of step (c), Holliday inherently cannot perform the remaining portion of step (c), namely, "sending with the update step or operation all transaction steps or operations for the transaction that have occurred prior to the update step or operation for the transaction, or prior to the previous update step or operation if a previous update step or operation existed for the transaction."

Nor do any of the other protocols in Holliday make up for the above-noted deficiencies in Protocol A3. In sum, Protocol A3 highlighted by the Examiner actually teaches away from step (c), and thus claim 6 is believed to be patentable over Holliday.

3. Patentability of claim 8 over Holliday

Claim 8 reads as follows (underlining added for emphasis):

8. A method of performing dual writes for replicating transactions among plural databases located at different nodes, each transaction being one or more transaction steps or transaction operations, at least some of the transaction steps or transaction operations being update steps or operations which are performed only after database locking occurs, the method comprising:
(a) initiating a transaction at an originating node; and
(b) the dual write replication process causing database locking to occur at one or more other nodes only upon the occurrence of an update step or operation in the transaction at the originating node.

Ideally, it is preferable to have access to all data elements in a database at all times. Placing locks on data elements detracts from this goal but is a necessary evil to avoid database synchronization problems, such as collisions.

Section X. 7.4-7.5 on pages 60-63 of the present specification describes one preferred embodiment of the present invention wherein read locks are propogated only if a data element is subsequently updated. Accordingly, there is no transmission or corresponding locks at the replicated nodes unless an update occurs. Since many read operations do not have a subsequent update operation, such as a write operation, this scheme reduces transmission overhead (i.e., there is a reduction in messages among nodes to perform a lock operation) and also reduces the number of read locks that have to be performed.

In the outstanding Office Action, the Examiner asserts that step (b) is also disclosed by Protocol A3 of Holliday. Applicants respectfully disagree. In Protocol A3, database locking occurs upon a write operation (e.g., update operation) but also occurs when there is a read operation. "Read locks" are explicitly discussed in Protocol A3. Nowhere does Holliday discuss any scheme wherein read locks only occur when there is a subsequent or corresponding write operation associated with the data element that is read. That is, the mere existence of write locks and read locks in Holliday is not a disclosure of the claim limitation wherein "database locking...occur[s] at one or more other nodes only upon the occurrence of an update step or operation in the transaction at the originating node." Protocol A3 of Holliday thus fails to disclose or suggest at least step (b) of claim 8.

Nor do any of the other protocols in Holliday make up for the above-noted deficiencies in Protocol A3. Claim 8 is thus believed to be patentable over Holliday.

4. Patentability of dependent claims

The dependent claims are believed to be allowable because they depend upon respective allowable independent claims, and because they recite additional patentable steps.

**Conclusion**

Insofar as the Examiner's rejections were fully addressed, the instant application is in condition for allowance. Issuance of a Notice of Allowability of all pending claims is therefore earnestly solicited.

Respectfully submitted,

BRUCE D. HOLENSTEIN et al.

March 23, 2006 By: Clark Jablon
(Date)

CLARK A. JABLON
Registration No. 35,039
AKIN GUMP STRAUSS HAUER & FELD LLP
One Commerce Square
2005 Market Street - Suite 2200
Philadelphia, PA 19103-7086
Telephone: (215) 965-1200
Direct Dial: (215) 965-1293
Facsimile: (215) 965-1210
E-Mail: cjablon@akingump.com

CAJ:gem